1-1-2015

Supervised machine learning model for high dimensional gene data in colon cancer detection

Huaming Chen
Lanzhou University, hc007@uowmail.edu.au

Hong Zhao
Xiamen University

Jun Shen
University of Wollongong, jshen@uow.edu.au

Rui Zhou
Lanzhou University

Qingguo Zhou
Lanzhou University, zhouqg@lzu.edu.cn

Supervised machine learning model for high dimensional gene data in colon cancer detection

Abstract

With well-developed methods in gene level data extraction, there are huge amount of gene expression data, including normal composition and abnormal ones. Therefore, mining gene expression data is currently an urgent research question, for detecting a corresponding pattern, such as cancer species, quickly and accurately. Since gene expression data classification problem has been widely studied accompanying with the development of gene technology, by far numerous methods, mainly neural network related, have been deployed in medical data analysis, which is mainly dealing with the high dimension and small quantity. A lot of research has been conducted on clustering approaches, extreme learning machine and so on. They are usuallly applied in a shallow neural network model. Recently deep learning has shown its power and good performance on high dimensional datasets. Unlike current popular deep neural network, we will continue to apply shallow neural network but develop an innovative algorithm for shallow neural network. In the supervised model, we demonstrate a shallow neural network model with a batch of parameters, and narrow its computational process into several positive parts, which process smoothly for a better result and finally achieve an optimal goal. It shows a stable and excellent result comparable to deep neural network. An analysis of thealgorithm is also presented in this paper.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Chen, H., Zhao, H., Shen, J., Zhou, R. & Zhou, Q. (2015). Supervised machine learning model for high dimensional gene data in colon cancer detection. IEEE BigData Congress (pp. 134-141). New York: IEEE.

Supervised Machine Learning Model for High Dimensional Gene Data in Colon Cancer Detection

Huaming Chen*, Hong Zhao†, Jun Shen‡, Rui Zhou* and Qingguo Zhou*
*School of Information Science and Engineering
Lanzhou University, Lanzhou, China
Email: chenhm12@lzu.edu.cn;zr@lzu.edu.cn;zhouqg@lzu.edu.cn
†Department of Physics
Xiamen University
Xiamen, China
Email: zhaoh@xmu.edu.cn
‡School of Information Systems and Technology
University of Wollongong
Wollongong, NSW, Australia
Email: jshen@uow.edu.au

***Abstract*—With well-developed methods in gene level data extraction, there are huge amount of gene expression data, including normal composition and abnormal ones. Therefore, mining gene expression data is currently an urgent research question, for detecting a corresponding pattern, such as cancer species, quickly and accurately. Since gene expression data classification problem has been widely studied accompanying with the development of gene technology, by far numerous methods, mainly neural network related, have been deployed in medical data analysis, which is mainly dealing with the high dimension and small quantity. A lot of research has been conducted on clustering approaches, extreme learning machine and so on. They are usuallly applied in a shallow neural network model. Recently deep learning has shown its power and good performance on high dimensional datasets. Unlike current popular deep neural network, we will continue to apply shallow neural network but develop an innovative algorithm for shallow neural network. In the supervised model, we demonstrate a shallow neural network model with a batch of parameters, and narrow its computational process into several positive parts, which process smoothly for a better result and finally achieve an optimal goal. It shows a stable and excellent result comparable to deep neural network. An analysis of the algorithm is also presented in this paper.**

***Keywords*-Neural Network; Monte Carlo; high dimensional data**

I. Introduction

Gene expression studies have been carried out for decades. They are obtained from multiple tissue samples, and by comparing the gene expression expressed in normal tissue with the one in diseased ones, we hope to obtain a better insight into the disease pathology [1]. These expression studies applied systematic approaches to understanding the correlation between gene expression profiles to disease states or different development stages of a cell [2]. The technological advancement in molecular biology, especially microarray analysis, facilitates new discoveries in basic research on biology and medicine. Hence quantitative measurement of gene expression plays an important role in finding clues about the mechanisms of gene regulation and interaction [3].

There is an important bond between a computational model and gene expression data by taking gene expression profiles and its corresponding disease states or stages of cells into consideration. Although several studies, such as [4], have demonstrated the potential performance of ensemble machine learning in classifying gene expression, it is still an chanllenge task to some extent.

As biologists expect, gene expression data obtained via high-throughput technologies vary in different experimental environments, at different developmental stages or in different tissues. Hence the data is organised in a matrix known as a gene expression profile [2]. In this way, this matrix always holds a high dimensionality. Because gene expression deviates from normal composition in case that a patient gets cancer [4], one of the goals is to determine the difference between gene expressions, including tumor cells and normal tissues, with that high-dimension data. So far, a number of computational approaches based on different data have been developed and employed to address the difference between gene expressions [5–8]. In particular, numerous machine learning classification techniques have been used for the classification of cancerous and normal tissues.

For gene expression data, its high dimensionality and a relative fewer quantity numbers require much more consideration and specific preprocessing to deal with. There are several methods to reduce the dimensionality from feature space into the hidden space, such as feature preselection with non-parametric scoring method of [9], principal component

Qingguo Zhou is the corresponding author

analysis(PCA) deployed in [1] and ensemble-based feature selection methods combined with the incremental feature selection(IFS) strategy in [10]. These research discussed about some potential ways to reduce dimensionality and drastically ease the computational burden.

However, to deal with these challenges, we would deploy in this paper a more general way in learning features from this original data and testing it with a valid data. We deploy a supervised machine learning model derivated from Monte Carlo methods, local field [11, 12] and support vector machine(SVM) theory. Compared with other methods and models mentioned above, via this machine learning model, it's been much easier to find patterns in high dimensional gene data and achieve a high accuracy.

The remainder of this paper is organized as follows: Section 2 provides a literature review of numerous machine learning classification methods; Section 3 outlines the supervised model and its core part; Section 4 describes results of our methods with comparisons with other methods, such as deep learning mentioned in [1]. Finally Section 5 concludes the paper.

II. Related Work

Previously several researches have shown that ensemble machine learning consistenly performs well in classifying biological data, and now that gene expression study has nowadays obtained some decent results, we could have a confirmed belief that with certain supervised and unsupervised machine learning techniques, the gene level classfication would be achieved.

In [2], considering these two types of learning schemes, namely supervised and unsupervised learning, Tan et.al. ensembled machine learning in classifying gene expression data on cancer classification problems. The conventional approach for such cancer classification is primarily based on the morphological appearance of the tumor. In this paper, they applied three major algorithms, single C4.5, Bagging and AdaBoost decision trees, to classify several gene expression datasets. However people would prefer herein is a common model and algorithm, which might become neural network rather than the learning schemes as mentioned in [2].

With regard to clustering approaches including hard and soft versions, Banu et.al. has been trying to mine gene expression to predict gene expression patterns and assist clinicians in early diagnosis of tumor formation [13]. Using clustering techniques from data mining perspective is trying to reveal the similarity between genes or a set of genes with similar conditions. This normally leads to a consequence list of expression patterns.

On the other hand, a lot of research have been focused on gene expressions such as gene expressions based colon classification scheme(GECC) as mentioned in [3]. In that paper, authors proposed a novel scheme for gene expressions based colon classification with mainly two complementary ways. One is reducing its overwhelmingly larger size to smaller ones via chi-square, F-Score and PCA. The other is a majority voting based on SVM to classify the given gene based samples. The authors assumed that formulation of gene expressions would remove the redundancy and fit itself better for a SVM classifier.

In consideration of high dimensionality, in [3] authors performed four class-imbalanced classifiers including DLDA(diagonal linear discriminant analysis), RFs(random forests), SVMs and SVM-based correction classifier. The latter SVM-based correction classifier is based on SVM threshold adjustment(SVM-THR). However the paper only focused on imbalanced class data.

The recent studies about deep learning show that it has a powerful and scalable performance on high dimensionality datasets. Even though currently unsupervised machine learning and deep learning are mainly used in image and audio processing domains, [1] included additional properties of the data, where certain capabilities have been extended to facilitate learning in a very high dimensional feature space. The authors of [1] used gene expression as an example. They try to implement unsupervised feature learning methods in gene expression analysis other than tumor architecture and images. In [1] PCA algorithm is deployed in the first phase. As for the linear transformation of PCA performed in the data, the authors use an unsupervised feature learning method in the second stage to facilitate a capture of nonlinearity of the relations between expressions of different genes, and to obtain more discriminating features. Furthermore, the second phase of feature learning approach is mainly focused on sparse autoencode. For this unspervised feature learning method, in which the input is used as the target for the output layer [14], it tries to learn a function $h(x) \approx x$ that represents an approximation of the input data. And in order to perform the task of cancer detectinon and cancer type classification, the features learned in the two stages adding with some random raw features from the input are subsequently used with a label dataset for classifier learning. Softmax regression is deployed as a learning classifier in that paper.

Through this structure described in [1], typically with three or four layers, it shows a relatively high feasibility and applicability for high dimensionality of data. However it is a complicated system which takes consideration of parameters and structures, such as transfer function and framework, which uses sparse autoencoder.

What we present in this paper is a novel supervised learning model, which could deal with high dimensional data with a small quantity. Transfer function and parameters presetting would be much less important in comparision with the cost function. However it could be a common model for every kinds of datasets. With a general three layers network and simple algorithm for adjusting these parameters

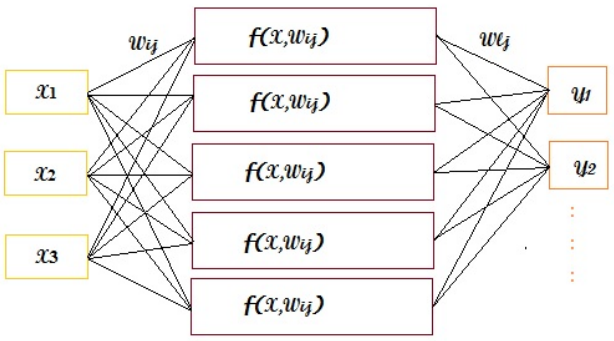

Figure 1. Basic Neural Network Diagram

according to the cost function, our results have shown a good perfomance and efficiency.

III. Algorithm

As for a normal neural network fitting into machine learning, it usually contains three layers: an input layer, a hidden layer and an output layer. This traditional model is shown as Figure. 1 in which $\{\boldsymbol{x}_i,\boldsymbol{y}_i\}$ serve as input layer units and output layer units individually while $\{\boldsymbol{W}_{ij},\boldsymbol{W}_{lj},\boldsymbol{h}_i,\boldsymbol{b}_i,\beta_i,\varepsilon,\boldsymbol{f}\}$ contribute for construction of hidden layer. In Figure. 1, $\boldsymbol{X}$ represents a set of $\boldsymbol{x}_i$. Here each layer is represented by three different groups of units which we consider as neurons. $\boldsymbol{W}_{ij}$ is the weight connection between input layer units $\boldsymbol{x}_i$ and hidden layer units $\boldsymbol{h}_i$. Through a transfer function $\boldsymbol{f}$, the output of hidden layer is defined by these parameter of each hidden layer units $\boldsymbol{b}_i,\beta_i$. Formula (1) and (2) shows the exact working flow formula for this Figure. 1. The parameter named ε is one adjustment amount for training a normal neural network.

$$\boldsymbol{h}_i = \boldsymbol{f}(\beta_i(\sum_{1}^{n} \boldsymbol{W}_{ij}*\boldsymbol{x}_i\text{-}\boldsymbol{b}_i)) \tag{1}$$

$$\boldsymbol{y}_i = \sum_{1}^{n} \boldsymbol{W}_{lj}*\boldsymbol{h}_j \tag{2}$$

There are two critical methods for designing such structure: back-propagation(BP) algorithm [15] and support vector machine(SVM) method.

Derivated from statistics theories, Monte Carlo Algorithm is previous considered as a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results [16]. Because it is mainly used in three distinct problem classes: optimization, numerical integration, and generation of draws from a probability distribution [11, 12], we slightly revise it and adapt it to our supervised model.

As support vector machine plays an important role in machine learning, mainly covering function fitting, pattern recognition and even time series prediction [17, 18], in this paper we employ Monte Carlo Algorithm in searching for general vectors rather than support vectors. The goal is the same, that is to say, mapping data, as much and precise as we can, into this machine.

To achieve general vectors in this shallow network rather than a deep network with multi layers more than a single hidden layer, we skip the support vector and search these parameter directly by Monte Carlo Algorithm. As support vectors always act a good performance for its structure, Monte Carlo Algorithm deployed in this shallow network leads to a better performance step by step. And within a praticable accuracy and time consumption, this structure would finally achieve a perfect result for data.

All the parameters inside this structure could be adapted to different circumstances. To illustrate it, we fix parameters $\boldsymbol{W}_{lj}$ between hidden layer and output layer. Within Monte Carlo Algorithm procedure, each time it would only change only one parameter from one same group if its cost function Δ shown as Formula. (3) leads to a better performance

Since this algorithm is highly flexible and it can be directly applied with different transfer functions and cost functions, we propose this shallow neural network structure designed by the revised Monte Carlo algotirhm.

A brief process of Monte Carlo Algorithm is shown as below. The entire parameters sets includes $\{\boldsymbol{W}_{ij},\boldsymbol{W}_{lj},\boldsymbol{h}_i,\boldsymbol{b}_i,\beta_i,\varepsilon,\boldsymbol{y}_i\}$. The neuron transfer function is presented as $\boldsymbol{f}$.

In this algorithm $\boldsymbol{W}_{ij}$ still represents the matrix between *input layer* and *hidden layer*. $\boldsymbol{W}_{lj}$ stands for the matrix between *hidden layer* and *output layer*. These two parameters are connection weights between two connected layer. $\boldsymbol{h}_i$ is the relative output of each hidden layer units. And $\boldsymbol{b}_i$ represents the bias of each hidden layer units. The β_i shows the transfer function coefficient. ε is a correcting unit for each parameters used in Monte Carlo Alogrithm. And finally $\boldsymbol{y}_i$ represents the output layer units. This shallow neural network uses ε to adjust all the other parameters to fit data features. Each round a random parameter is selected and changed to a new value within its available interval. When it leads to a better result where we call it cost function Δ, which means Δ reduces or holds, we accept this adjustment.

A. Set parameters

Set these parameters $W_{ij}, W_{lj}, h_i, b_i, \beta_i, \varepsilon, y_i$ to a random value within a fixed interval. This interval would be different due to its role in this structure.

B. Change parameters

Choose one group of the parameters from $\{\boldsymbol{W}_{ij},\boldsymbol{W}_{lj},\boldsymbol{b}_i,\beta_i\}$.

Randomly choose one parameter from this group, adjust

it with the ε and calculate $\boldsymbol{y}_i$ and Δ. If Δ is not becoming worse, then we accept this change and move on to next round. This step stops after a Monte Carlo Steps.

C. Repeat of above step to other parameters

Change to other types of parameters from $\{\boldsymbol{W}_{ij}, \boldsymbol{W}_{lj}, \boldsymbol{b}_i, \beta_i\}$ after a Monte Carlo Steps.

D. Stop training

Repeatedly train this shallow neural network via B and C steps and stop when either a critical time runs out $t \geq t_0$ or cost function $\Delta \leq \Delta_0$.

This shallow neural network would can be applied practically to different scenarios such as function fitting, pattern recognition and time series prediction etc. Here we apply it with a high dimensional data which was used in [1] and [2]'s paper, the Colon Cancer data with 2000 dimensions.

We demonstrate the Colon Cancer classification system with a cost function shown as below (3).

$$\Delta = \sqrt{\frac{1}{n}\sum_{1}^{n}(y_i - y_{i0})^2} \tag{3}$$

In Colon Cancer dataset, there are two main consequences. One is Benign, which means non-cancer, the other is Malignant standing for cancer patients. Since this dataset's sample number is low, its dimension is extremly high compared to its number. We adopt the strategy shown in [1], 32 sample for training and 30 sample for testing. For pattern recognition, in this case, the output layer has two neurons. And further we set a separating margin d for Δ. It's shown as (4).

$$\Delta = \sum_{1}^{n}((y_{i1} * y_{i10} - d)^2 + (y_{i2} * y_{i20} - d)^2) \tag{4}$$

Here y_{i1} and y_{i2} reprensent the output layer units, y_{i10} and y_{i20} represent relative labels of dataset. When consequence comes to Benign, y_{i10} and y_{i20} are $(+1, -1)$. Otherwise they are $(-1, +1)$. Since the Monte Carlo Algorithm in this supervised learning machine aims to get a local field separation, which indeed is the state of these two output layer neurons, there is a precondition: in case of $y_i * y_{i0} > d$, Δ doesn't count this sample. Through this method, we could separate samples into a reasonable margin and get rid of an over learning condition.

In this Monte Carlo algorithm, cost function Δ is the critical factor. For different systems, it varies to fit the practical problem such as function fitting, pattern recognition and time series prediction. By designing different cost function, we could fit it to different situation perfectly. In next section we will discuss about Colon Cancer's result.

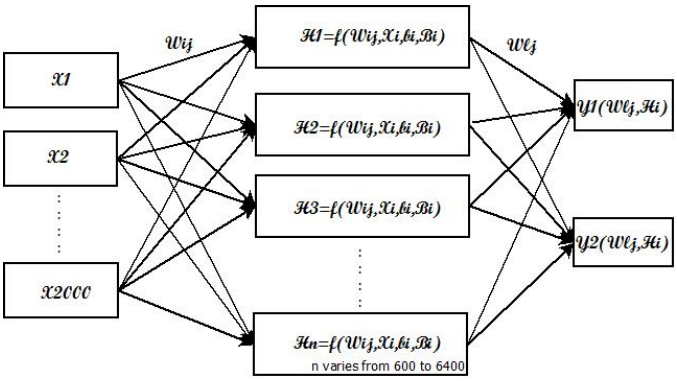

Figure 2. Deployed Neural Network Model

IV. Experiments & Result

Different from other methods such as deep learning, SVM, there is not pre-processing with datasets. In certain circumstances the noise, which is inevitably existing in datasets, becomes the main problems for models to deal with. With pre-processing, these original methods suppose they would get a better dataset without noise and process it for better performance as a result.

However, noises differs from each other and herein we want to evacuate it without any pre-processing. Actually within the local field theory according to [11, 12, 19], noise could be part of this model and facilitate this model's building. In that case, we directly input the dataset into this shallow neural network. Hence we get an input layer with certain units, and accordingly the output layer is multi- or single units presenting different consequences. Cost function would be different for different datasets but finally it would fits well to these datasets if we simply seperated this high dimensional datasets into classification task and time series predicting task.

A. Dataset

As we presented, we are using the Colon Cancer dataset from [1, 2]. With that datasets, the input layer of this supervised learning model deployed in Monte Carlo Algorithm would be 2000 units and its outputs are two units standing for normal and abnormal symptoms. The hidden layer varies in neural numbers as shown in Figure. 2.

B. Model

As shown in Figure. 2, this is the original shallow neural network with 2000 units input, 2 units output and hidden neural varies from 600 to 6400.

C. Procedure & Pseudocode

Since there are lots of parameters positively changing for a better performance, which means a lower cost function

here, we chose W_{ij}, β_i, b_i as adjustment factors marked in Figure. 2.

Initialization:

Set $W_{ij}, W_{lj}, h_i, b_i, \beta_i, \varepsilon, y_{io}$ to its interval values.
For W_{lj}, it is randomly set by +1 or -1 for simplifying the model. And here h_i is 0 in initialization step while y_{io} stands for corresponding correct state with (+1,-1) or (-1,+1). And ε is set to ε_0

Input:

The original Calon Cancer Datasets has 62 units with 2000 dimension and is the direct input for supervised model. In this experiment, 32 units serve as training dataset while the rest 30 units are used as a test dataset.

Output:

In this model, a well-functional output layer is designed accompanied with a cost function. y_{io} is the originally state for 62 units. And it is used in two situations, one is for calculating the cost function which is positively decreased by steps, and the other is a valid state to evaluate the model's performance.

Cost Function:

The key role of cost function in the supervised model would facilitate its performance and efficiency. In this paper we present a classification function, which is actually pattern recognition. So the cost function in this model is shown as Formula. (4).

It is much simpler in this experiment with an output layer is simplified within (+1,-1) and (-1,+1). Δ is positively decreased in Monte Carlo Algorithm process and finally approximate zero. Here we set Δ_0 equals 10^{-6}.

As mentioned in previous section, without pre-processing, we set the interval of W_{ij}, β_i to (-0.1,+0.1). The interval of b_i is (-1.0,+1.0). The deep learning method tries to locate a better set of initial arguments for training and adjusting. It could finally achieve a good and efficient training process. A proper interval of arguments would contribute a lot for Monte Carlo Algorithm, that is what we want to discuss more in future work.

We fixed W_{lj} to either +1 or -1. It is a random setting. Because in this supervised model, these two parameters W_{lj} and W_{ij} serve as two mirrors from mapping sample data to feature space, and eventually output space. It shows that regardless of W_{lj}, by adjusting W_{ij}, b_i, β_i, the mapping from data to output space achieves a good performance. And the transfer neuron function would be different in this system, however we only apply *tanh* function as it acts well.

Algorithm 1 Routing algorithm

init supervised model: Calculate Output Layer status y_i with a pre-setting values of all parameters.
Calculate Cost Function: Calculate Cost Function value Δ_1.

if $\Delta_1 \geq \Delta_0$ **then**
  if $step < Monte_Carlo_step$ **then**
   ε = random(+ε_0,-ε_0);
   Select a random W_ij from W_ij matrix;
   W_ij = W_ij + ε;
   update Output Layer status y_i;
   Calculate Cost Function value Δ;
  while $\Delta \leq \Delta_0$ **:**
    update W_ij;
    update Δ_1;
  end if

  if $step < Monte_Carlo_step$ **then**
   ε = random(+ε_0,-ε_0);
   Select a random β_i from β_i matrix;
   β_i = β_i + ε;
   update Output Layer status y_i;
   Calculate Cost Function value Δ;
  while $\Delta \leq \Delta_0$ **:**
    update β_i;
    update Δ_1;
  end if

  if $step < Monte_Carlo_step$ **then**
   ε = random(+ε_0,-ε_0);
   Select a random W_ij from W_ij matrix;
   b_i = b_i + ε;
   update Output Layer status y_i;
   Calculate Cost Function value Δ;
  while $\Delta \leq \Delta_0$ **:**
    update b_i;
    update Δ_1;
  end if
end if

According to local field theory, we set a prerequisites for cost function calculation in the supervised model. For y_i beyond +d or -d, it is dismissed for summing up in Formula. (4). And this Δ needs to be calculated and updated everytime when any of W_{ij}, β_i, b_i is changed. With this method, we avoid the risk of over-fitting, and in order to make supervised model learning from datasets as much as it can, cost function Δ_0 is diverse according to different situations.

In Figure. (3), we set n=600 and its d value increased

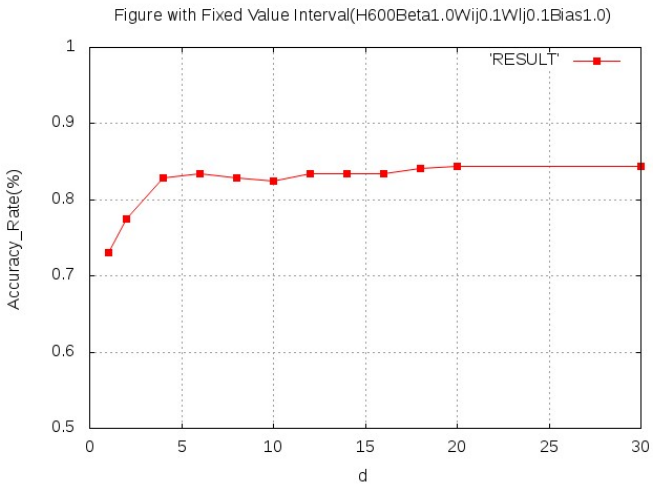

Figure 3. Separate Margin(means d value) Increases Results

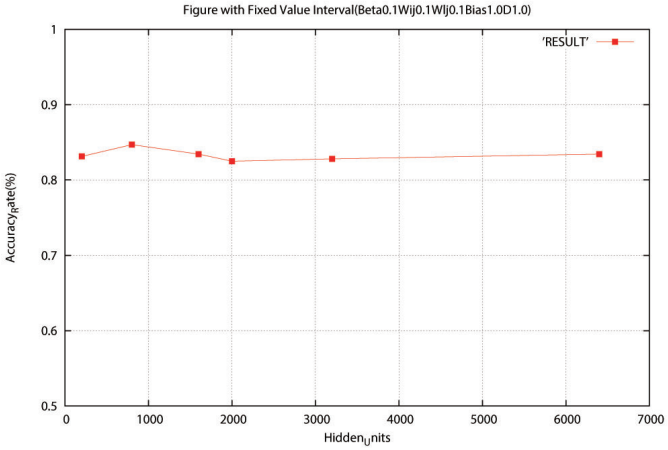

Figure 4. Hidden Unit Increases Results

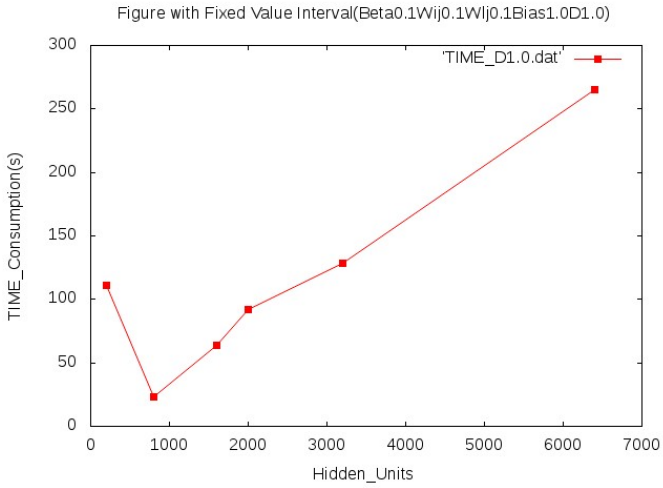

Figure 5. Hidden Unit Increases Time Consumption

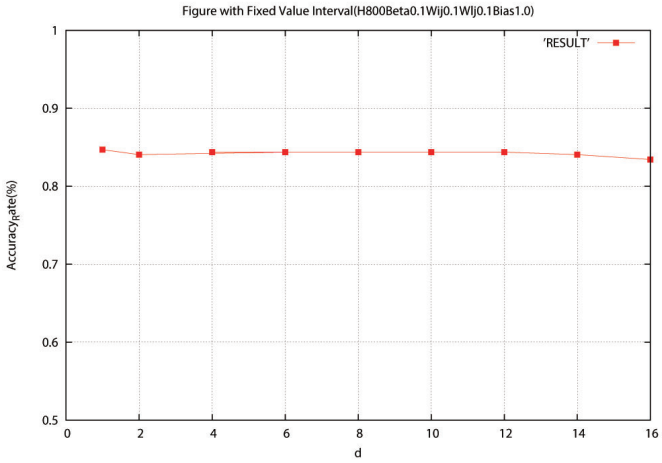

Figure 6. Separate Margin Increases Results

slightly. Therefore we get a well defined curve showing a joint powerful function by d and cost function.

As shown in Figure. (4), we fix d value to 1. In this certain saparate margin, we increase hidden units number. The accuracy keeps a relatively high value as demonstrated in Figure. (4). And the results are actually the same with the result with a deep learning structure as shown in [1].

What we consider here contains not only accuracy achieved but also time consumed. As shown in Figure. (5) a time consumption curve demonstrate exactly how much time is consumed in the condition that hidden units increases. These results is based on an average value of ten times tests for each dividuals.

With experiments, we find that when hidden layer contains 800 units, the system achieve a best performance with parameters's interval fluctuations. As shown in Figure.

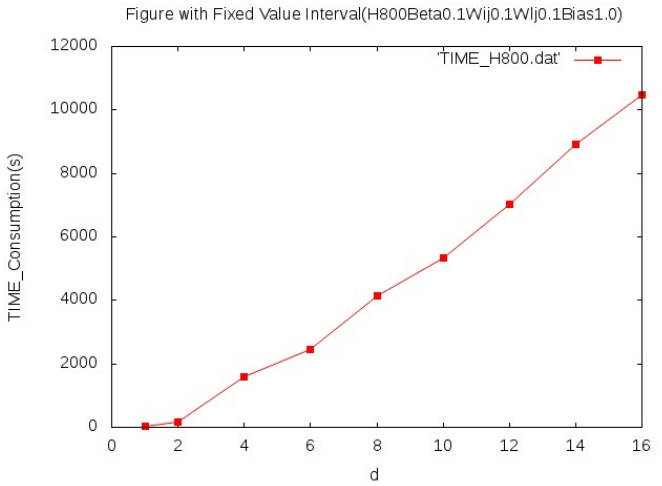

Figure 7. Separate Margin Increases TIME Consumption

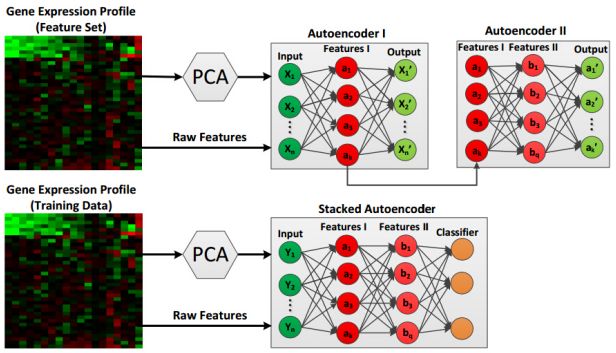

Figure 8. Deep Learning Structure from [1]

(6), d value is increasing while hidden layer holds a 800 units number. The value of d increases from 1 to 18 and the accuracy result stays around 84%.

As shown in Figure. (7), for a certain 800 units hidden layer, time consumption also varies from each other. However we discovered that it was much more efficient than other methods or algorithm mentioned above.

V. Discussions

As shown in Figure. (8), authors in [1] tried to find a general approach for cancer detection from gene expression datasets. The proposed approach included feature learning and classifier learning. Authors demonstrated that high dimensionality(on the order of 2000 to 50000) in gene expression data was reduced by PCA. But considering PCA only performing a linear transformation on the data, the resulting extracted features are simply a linear function of the original input data. And for non-linear relations, the authors randomly added some of the original raw features to the PCA features to form an augmented data for unsupervised feature learning [1]. Hence there was a barely vague understanding about the real relationship between data and labels.

Instead of using PCA for a linearity dimensionality reducing, we directly input the original raw data into the supervised model. According to universal approximation theorem, an arbitrarily random function could be obtained by a single hidden layer network with a finite number of neurons. Here by applying *tanh* function in the hidden layer as transfer function, we would extract the inside relations between datas from both non-linearity and linearity aspects. It is much more general to reproduce and apply it in different datasets. Monte Carlo algorithm deployed in the supervised model indeed facilitates this process, while cost function is an important part for stablizing the structure and achieving a good performance.

As storage and analysis of vast amounts of biological data is growing in the current booming bioinformatics development, there is an obvious gap between sequencing throughput and computer capbilities in dealing with such big data [20], typically handling data sets that approximate 5 Mb per sequence in DNA microarray [21]. What bioinformatics want to obtain here is capturing those elusive bonds between different datasets and the corresponding states. We have now seen an embarrassing situation that the cost to produce data is falling dramatically, which means much more different datasets from different sequences are showing up, while the efficiency of analysis remains at a low level [22].

Thus, a stable and efficient computational method for this situation is desired. Through our current experiment, as shown in Figures (3), (4), (6), we have obtained some results in Colon Cancer dataset. Compared with the performance in [1], our algorithm achieves a stable and efficient results. It can also be extended to other datasets and to find much more bonds between gene expressions, including time series prediction.

With the idea derivated from Hadoop [20] and Monte Carlo algorithm mentioned in the above sections, multi supervised models deployed on different datasets are under consideration for future work in order to explore elusive bonds between these datasets.

VI. Conclusion

In this paper, we demonstrate a pratically simple shallow neural network to accomplish the supervised machine learning model on gene expression data. Deployed with a Monte Carlo algorithm, we could control the whole system moving positively towards a better result. And a well-considered design of cost function plays a key role in minimizing the risk of over learning and makes the system to achieve a result as good as deep learning.

The Monte Carlo Algorithm is realized mainly with a special strategy. Every time only one parameter would probably be changed and thus the relative output could be updated in a positive and efficient way, due to some tricks from matrix operation. In that way, the time consumption is under control and the pratical performance could be achieved.

So far we achieve a good result with a single supervised model on colon cancer dataset. With a voting strategy, it could be expanded to multi supervised model with Monte Carlo algorithm for a same problem and a better performance is in sight.

With the beneficial strategy of Monte Carlo Algorithm, we search for the best control parameter set rather than a best machine. However, as the datasets have been growing in a dramatical speed both in number and size, bioinformatics is being considered a critical big data area. What big data aims not only to some real large dataset, but also to many other aspects [23, 24]. But what all the original algorithms or new emerging technologies could achieved on these gene expression datasets are still under

considerations and experiments. In future we will conduct this supervised machine learning model in more datasets, as well as time series prediction and function fitting areas.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China under Grant No. 61402210 and 60973137, Program for New Century Excellent Talents in University under Grant No. NCET-12-0250, "Strategic Priority Research Program" of the Chinese Academy of Sciences with Grant No. XDA03030100, Gansu Sci.&Tech. Program under Grant No. 1104GKCA049, 1204GKCA061 and 1304GKCA018. The Fundamental Research Funds for the Central Universities under Grant No. lzujbky-2014-49, lzujbky-2013-k05, lzujbky-2013-43, lzujbky-2013-44, lzujbky-2014-235 and lzujbky-2012-44, Gansu Telecom Cuiying Research Fund under Grant No. lzudxcy-2013-4, Google Research Awards and Google Faculty Award, China.

REFERENCES

[1] R. Fakoor, F. Ladhak, and A. N. et al., "Using deep learning to enhance cancer diagnosis and classification," in *Proceedings of the ICML Workshop on the Role of Machine Learning in Transforming Healthcare (WHEALTH). Atlanta, GA*, 2013.

[2] A. C. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification," *Applied Bioinformatics*, vol. 2, no. 3, pp. 75–83, 2003.

[3] W. J. Lin and J. J. Chen, "Class-imbalanced classifiers for high-dimensional data," *Class-imbalanced classifiers for high-dimensional data*, vol. 14, no. 1, pp. 13–26, 2013.

[4] S. Rathore, M. Hussain, and A. Khan, "Gecc: Gene expression based ensemble classification of colon samples," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, pp. 1131–1145, 2014.

[5] F. Li, C. Li, M. Wang, and et al., "Glycomine: a machine learning-based approach for predicting n-, c- and o-linked glycosylation in the human proteome," *Bioinformatics*, 2015.

[6] M. K. K. Leung, H. Y. Xiong, and L. J. L. et al., "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. 121–129, 2014.

[7] S. L. Win, Z. Z. Htike, and F. Yusof, "Cancer recurrence prediction using machine learning," *International Journal of Computational Science and Information Technology*, vol. 6, no. 1, 2014.

[8] S. S. Wei, H. J. Lu, Y. Lu, and et al., "An improved weight optimization and cholesky decomposition based regularized extreme learning machine for gene expression data classification," *Extreme Learning Machines 2013: Algorithms and Applications.*, pp. 55–66, 2014.

[9] P. J. Park, M. Pagano, and M. Bonetti, "A nonparametric scoring algorithm for identifying informative genes from microarray data," 2000, pp. 52–63.

[10] Z. Cai, D. Xu, Q. Zhang, and et al., "Classification of lung cancer using ensemble-based feature selection and machine learning methods," *Mol. BioSyst.*, vol. 11, no. 3, pp. 791–800, 2015.

[11] H. Zhao and T. Jin, "A global algorithm for training multilayer neural networks," *arXiv preprint physics*, 2006.

[12] H. Zhao, "Designing asymmetric neural networks with associative memory," *Physical Review E*, vol. 70, no. 6, 2004.

[13] P. K. N. Banu and S. Andrews, "Performance analysis of hard and soft clustering approaches for gene expression data," *International Journal of Rough Sets and Data Analysis*, vol. 2, no. 1, pp. 58–69, 2015.

[14] Ng and A.Y., "Unsupervised feature learning and deep learning @online." [Online]. Available: http://ufldl.stanford.edu/

[15] M. Patel, D. Mehta, P. Paterson, and R. Rawal, "An introduction to back propagation learning and its application in classification of genome data sequence," in *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*, ser. Advances in Intelligent Systems and Computing, 2014, vol. 236, pp. 609–615.

[16] Wikipedia. Monte carlo method. [Online]. Available: http://en.wikipedia.org/wiki/Monte_Carlo_method

[17] U. Thissen, R. V. Brakel, A. P. D. Weijer, and et al., "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35–49, 2003.

[18] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, 2009.

[19] H. Zhao and T. Jin, "Pattern recognition using asymmetric attractor neural networks," *Physical Review E*, vol. 72, no. 6, 2005.

[20] L. Dai, X. Gao, Y. Guo, and et al., "Bioinformatics clouds for big data manipulation."

[21] J. Rozas, J. C. Sanchez-DelBarrio, X. Messeguer, and et al, "Dnasp, dna polymorphism analyses by the coalescent and other methods," *Bioinformatics*, vol. 19, no. 18, pp. 2496–2497, 2003.

[22] M. E. Winn, "Big data challenges in bioinformatics the collision of technology and biology," Tech. Rep. 11, 2014.

[23] L. Wang and J. Shen, "Economical data-intensive service provision supported with a modified genetic algorithm," *2013 IEEE International Congress on Big Data (BigData Congress 2013)*, 2013.

[24] C. K. Mayer-Schnberger V, *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.